# Classifying Regular Languages via Cascade Products of Automata

Marcus Gelderie

RWTH Aachen, Lehrstuhl für Informatik 7, D-52056 Aachen
gelderie@automata.rwth-aachen.de

**Abstract.** Building on the celebrated Krohn-Rhodes Theorem we characterize classes of regular languages in terms of the cascade decompositions of minimal DFA of languages in those classes. More precisely we provide characterizations for the classes of piecewise testable languages and commutative languages. To this end we use biased resets, which are resets in the classical sense, that can change their state at most once. Next, we introduce the concept of the scope of a cascade product of reset automata in order to capture a notion of locality inside a cascade product and show that there exist constant bounds on the scope for certain classes of languages. Finally we investigate the impact of biased resets in a product of resets on the dot-depth of languages recognized by this product. This investigation allows us to refine an upper bound on the dot-depth of a language, given by Cohen and Brzozowski.

## 1 Introduction

A significant result in the structure theory of regular languages is the Krohn-Rhodes Theorem [7], which states that any finite automaton can be decomposed into simple "prime factors" (a detailed exposition is given in [4, 6, 9, 10]).

We use the Krohn-Rhodes Theorem to characterize classes of regular languages in terms of the decompositions of the corresponding minimal automata. In [8] this has been done for star-free languages by giving an alternative proof for the famous Schützenberger Theorem [11]. In [1] $\mathcal{R}$-trivial languages are characterized (among other things) by proving structural properties of the cascade products covering their minimal automata. We continue these studies, in an attempt to improve our understanding of the potential of automata decompositions for classifying regular languages, an approach which is as yet not well developed in comparison to the structure theory of regular languages based on algebraic methods (wreath product and block product decomposition of monoids, see [14]).

We treat here the case of piecewise testable and commutative languages, as well as the star-free languages in their classification by the dot-depth hierarchy. To this end, we use the concept of a *biased reset* (called a *half reset* in [1]) and introduce *locally i-triggered cascade products* in order to characterize piecewise testable languages. For commutative languages we use the notion of one letter automaton (OLA) and a corresponding one letter cascade product. We show that a language is commutative iff its minimal automaton is covered by a direct product of a one letter cascade product of biased resets and one letter simple cyclic grouplike automata.

Next we introduce the notion of *scope* of resets within a cascade product in order to further refine our analysis of the Krohn-Rhodes decomposition. Informally speaking, the scope of a product of resets is the maximal number of preceding automata, to which any given factor is still sensitive. The scope measures a notion of locality in the product. As initial results we show that the scope of cascade products recognizing $\mathcal{R}$-trivial languages is bounded by 2 and that for piecewise testable languages, it is bounded by 1.

Finally we pick up a result from Cohen and Brzozowski [2], which bounds the dot-depth of a star-free language by the number of factors of a cascade product of resets recognizing it. We show that this result can be refined by counting blocks of biased resets as a single reset. To this end we show that multiplying (w.r.t. the cascade product) an arbitrary number of biased resets to an automaton $\mathfrak{A}$ increases the dot-depth of languages recognized by the product by at most one (compared to the dot-depth of languages already recognized by $\mathfrak{A}$).

The present paper is based on the diploma thesis [5].

## 2 Preliminaries

A *semiautomaton* is a tuple $\mathfrak{A} = (Q, \Gamma, \delta^{\mathfrak{A}})$, where $Q$ is a finite set of *states*, $\Gamma$ is a finite set of *letters*, called the *input alphabet of* $\mathfrak{A}$ and $\delta^{\mathfrak{A}} : \Gamma \to Q^Q$ is the *state transition function* assigning a mapping $\overline{a}^{\mathfrak{A}} : Q \to Q$ to each letter $a \in \Gamma$. By function composition we can extend these mappings to words $w = a_1 \cdots a_n \in \Gamma^*$ by setting $\overline{w}^{\mathfrak{A}}(q) = \overline{a_n}^{\mathfrak{A}}(\overline{a_{n-1}}^{\mathfrak{A}}(\cdots \overline{a_1}^{\mathfrak{A}}(q)\cdots))$[1]. A *subsemiautomaton* of $\mathfrak{A}$ is a structure $\mathfrak{A}_0 = (Q_0, \Gamma, \delta_0^{\mathfrak{A}})$, where $Q_0 \subseteq Q$ is closed under the mappings $\delta^{\mathfrak{A}}(a)$ for all $a \in \Gamma$ and $\delta_0^{\mathfrak{A}}(a)$ is the restriction of $\delta^{\mathfrak{A}}(a)$ to $Q_0$, $a \in \Gamma$. A *homomorphism* from $\mathfrak{A} = (Q, \Gamma, \delta^{\mathfrak{A}})$ to $\mathfrak{B} = (P, \Gamma, \delta^{\mathfrak{B}})$ is a mapping $\varphi : Q \to P$ with $\varphi(\overline{a}^{\mathfrak{A}}(q)) = \overline{a}^{\mathfrak{B}}(\varphi(q))$ for all $q \in Q$ and $a \in \Gamma$. $\mathfrak{A}$ *covers* a semiautomaton $\mathfrak{B}$ (of the same input alphabet), written $\mathfrak{B} \leq \mathfrak{A}$, if $\mathfrak{B}$ is the image of a subsemiautomaton of $\mathfrak{A}$ under some homomorphism $\varphi$.

A *deterministic finite automaton (DFA)* is a semiautomaton $\mathfrak{A} = (Q, \Gamma, \delta)$ with a designated *initial state* $q_0 \in Q$ and a set $F \subseteq Q$ of *final states.* In this situation we sometimes write $(\mathfrak{A}, q_0, F)$ or (using the same symbol for the DFA and the corresponding semiautomaton) $\mathfrak{A} = (Q, \Gamma, q_0, \delta, F)$. If $\mathfrak{A}$ is a DFA, then the *language accepted by* $\mathfrak{A}$ is denoted by $L(\mathfrak{A}) = \{w \in \Gamma^* | \overline{w}^{\mathfrak{A}}(q_0) \in F\}$. A single semiautomaton constitutes the foundation of several DFA. Hence, a semiautomaton "recognizes" a set of languages: $\mathcal{L}(\mathfrak{A}) = \{L \subseteq \Gamma^* | \exists q_0 \in Q,\ F \subseteq Q :\ L = L(\mathfrak{A}, q_0, F)\}$. If $L \in \mathcal{L}(\mathfrak{A})$ we say $L$ is *recognized* by $\mathfrak{A}$. Given a regular language $L \subseteq \Gamma^*$ we denote the *canonical DFA for* $L$ by $\mathfrak{A}_L$. As noted above, we often identify a DFA with the underlying semiautomaton. This identification is used in the following proposition, the proof of which is left to the reader:

[1] Note that the empty word induces the identity mapping.

**Proposition 1.** *Given a semiautomaton* $\mathfrak{A}$ *and a regular language* $L$*, we have* $L \in \mathcal{L}(\mathfrak{A})$ *iff* $\mathfrak{A}_L \leq \mathfrak{A}$*.*

In what follows we will often deal with two kinds of semiautomata: *resets* and *permutation* automata. A reset automaton[2] is a semiautomaton $\mathfrak{R} = (\{0,1\}, \Gamma, \delta^{\mathfrak{R}})$, where for any input $a \in \Gamma$ the induced mapping $\overline{a}^{\mathfrak{R}}$ is either the identity on $\mathbb{B} := \{0,1\}$ or has constant value $x \in \mathbb{B}$. Conversely, a permutation automaton is an automaton $\mathfrak{P} = (Q, \Gamma, \delta^{\mathfrak{P}})$, such that every input $a \in \Gamma$ induces a permutation (that is, a bijective function).

Recall that a monoid $M$ *divides* a monoid $N$, written $M \prec N$, if there exists a surjective monoid homomorphism $\psi : N_0 \to M$ from a submonoid $N_0$ of $N$ onto $M$. As with coverings of automata, the division relation is transitive and reflexive. We recall that the transition monoid of an automaton is the set $M(\mathfrak{A})$ of all mappings $\overline{w}^{\mathfrak{A}} : Q \to Q$ for $w \in \Gamma^*$. In the special case where $\mathfrak{A}$ is a permutation automaton, the monoid $M(\mathfrak{A})$ is a group $G$. If $\mathfrak{A}$ has precisely $|G|$ states we call $\mathfrak{A}$ a *grouplike automaton*. Notice that in this case, we may identify the states from $Q$ with elements from $G$. $G$ is the group *associated* with $\mathfrak{A}$. A grouplike automaton $\mathfrak{G}$ is simple (cyclic) if the associated group is simple (cyclic). Since the number of states is equal to $|G|$ it makes sense to speak of the *order of* $\mathfrak{G}$, which we define to be the order of $G$.

Given two automata $\mathfrak{A} = (Q, \Gamma, \delta^{\mathfrak{A}})$ and $\mathfrak{B} = (P, \Gamma \times Q, \delta^{\mathfrak{B}})$ we define

$$\begin{aligned} \delta^{\mathfrak{A}*\mathfrak{B}}(a) := \overline{a}^{\mathfrak{A}*\mathfrak{B}} &: Q \times P \to Q \times P \\ (q,p) &\mapsto (\overline{a}^{\mathfrak{A}}(q), \overline{(a,q)}^{\mathfrak{B}}(p)) \end{aligned}$$

The automaton $\mathfrak{A} * \mathfrak{B} = (Q \times P, \Gamma, \delta^{\mathfrak{A}*\mathfrak{B}})$ is called the *cascade product of* $\mathfrak{A}$ *and* $\mathfrak{B}$. We recall a few important properties of the cascade product:

**Theorem 1 (see [6]).** *Let* $\mathfrak{A}$*,* $\mathfrak{B}$ *and* $\mathfrak{C}$ *be semiautomata with input alphabets of the suitable format. Then the following hold:*

*(1)* $(\mathfrak{A} * \mathfrak{B}) * \mathfrak{C} = \mathfrak{A} * (\mathfrak{B} * \mathfrak{C})$
*(2) If* $\mathfrak{A} \leq \mathfrak{B}$ *then* $\mathfrak{C} * \mathfrak{A} \leq \mathfrak{C} * \mathfrak{B}$

The following theorem is the basis for our task of characterizing language classes:

**Theorem 2 (Krohn, Rhodes, [7]).** *Let* $\mathfrak{A}$ *be a semiautomaton. Then*

$$\mathfrak{A} \leq \mathfrak{F}_1 * \cdots * \mathfrak{F}_n$$

*for semiautomata* $\mathfrak{F}_i$*, such that each* $\mathfrak{F}_i$ *is either a reset or a simple grouplike automaton with* $M(\mathfrak{F}_i) \prec M(\mathfrak{A})$*.*

A detailed exposition of the Krohn-Rhodes Theorem is given in [3, 4, 6]. We will call a decomposition of an automaton $\mathfrak{A}$, which is of the form stated in Theorem 2, a *Krohn-Rhodes decomposition (of* $\mathfrak{A}$*).*

---

[2] Reset automata are often introduced in a more general fashion, allowing for an arbitrary number of states. However, if a reset automaton has more than 2 states, it can be covered by a direct product of 2-state reset automata (see, for instance, [6]).

## 3 Piecewise Testable Languages

In this section we want to characterize piecewise testable languages in terms of their cascade products. Recall that a language is *piecewise testable* iff it is a Boolean combination of expressions of the form $\Gamma^* a_1 \Gamma^* \cdots \Gamma^* a_n \Gamma^*$ for letters $a_1, \ldots, a_n \in \Gamma$, $n \in \mathbb{N}_0$ (if $n = 0$ we obtain $\Gamma^*$). There are several characterizations for piecewise testable languages (see for instance [1, 9, 10, 12, 15, 13]). It should be mentioned that the characterization from [13] (stated in terms of matrices over a semiring) is very similar to the one we give below, yet not stated in terms of semidirect products. This result is again mentioned in [15] in the context of a discussion of restricted semidirect products. The decomposition obtained thereby is indeed very close the one we give below. As we give a purely automaton theoretic proof (the proof from [15] is purely algebraic) it would be interesting to investigate whether one can be obtained from the other.

**Definition 1.** *Let $\mathfrak{R}_1 * \cdots * \mathfrak{R}_n$ be a cascade product of resets.*

*(a) We say the reset $\mathfrak{R}_i$ has* scope $k$ *if for every pair of inputs $(a, (x_1, \ldots, x_{i-1}))$ and $(a, (y_1, \ldots, y_{i-1}))$ with $y_j = x_j$ for all $i - k \leq j \leq i - 1$ the induced mappings are equal, i.e.*

$$\overline{(a, (x_1, \ldots, x_{i-1}))}^{\mathfrak{R}_i} = \overline{(a, (y_1, \ldots, y_{i-1}))}^{\mathfrak{R}_i}$$

*(b) The* scope of $\mathfrak{R}_i$ *is the minimal number $k$, such that $\mathfrak{R}_i$ has scope $k$. The scope of the product $\mathfrak{R}_1 * \cdots * \mathfrak{R}_n$ is the maximal scope of a reset $\mathfrak{R}_i$.*

*(c) If $b \in \mathbb{B}$, we call the product above* strictly locally $b$-triggered*, if it has scope 1 and for every reset $\mathfrak{R}_i$ every input of the form $(a, (x_1, \ldots, x_{i-2}, 1 - b))$ induces the identity mapping. A product is called* locally $b$-triggered*, if it is a direct product of strictly locally $b$-triggered products.*

In other words, the scope of a reset counts the number of resets preceding it in a cascade, such that it is sensitive to the state of these resets. For example, a cascade product degenerates to a direct product iff it has scope 0. Strictly locally $b$-triggered products add another constraint to this, namely that the reset may only alter its state if the immediately preceding reset is in state $b$. We will return to the scope of a cascade product in Sec. 5.

A reset $\mathfrak{R} = (\mathbb{B}, \Gamma, \delta)$ is *$b$-biased*, where $b \in \mathbb{B}$, if for every input $a \in \Gamma$ we have $\overline{a}^{\mathfrak{R}} = \mathrm{id}_{\mathbb{B}}$ or for all $q \in \mathbb{B}$ we have $\overline{a}^{\mathfrak{R}}(q) = b$. In the second case, we write $\overline{a}^{\mathfrak{R}} \equiv b$. A reset $\mathfrak{R}$ is *biased* if it is $b$-biased for some $b$. We can now state:

**Theorem 3.** *A language $L$ is piecewise testable iff it is recognized by a locally $b$-triggered cascade product of $b$-biased resets.*

Notice that this yields a Krohn-Rhodes decomposition of $\mathfrak{A}_L$ by Proposition 1. We will dedicate the remainder of this section to proving this result. To this end we first verify that all piecewise testable languages are recognized by a locally $b$-triggered cascade product of $b$-biased resets. Without loss of generality we will henceforth assume $b = 1$ (otherwise we rename the states).

By the definition of piecewise testable languages above it is sufficient to show that every language of the form $L = \Gamma^* a_1 \Gamma^* \cdots \Gamma^* a_n \Gamma^*$ is recognized by such a product: Boolean combinations can be recognized by direct products and a direct product of locally 1-triggered products is again locally 1-triggered. Given $L$, we define $n$ 1-biased resets $\mathfrak{R}_1, \dots, \mathfrak{R}_n$ by setting[3]:

$$\overline{(a,(x_1,\dots,x_{i-1}))}^{\mathfrak{R}_i}(b) = \begin{cases} 1 \text{ if } x_{i-1} = 1 \wedge a = a_i \\ b \text{ otherwise} \end{cases}$$

for all $b \in \mathbb{B}$, $a \in \Gamma$ and $(x_1,\dots,x_{i-1}) \in \mathbb{B}^{i-1}$. Then every $\mathfrak{R}_i$ is 1-biased and $\mathfrak{R}_1 * \cdots * \mathfrak{R}_n$ is strictly locally 1-triggered and evidently accepts $L$ with initial state $q_0 = (0,\dots,0)$ and final states $\{(1,\dots,1)\}$.

Before embarking on showing the converse, we need a bit of preparation:

*Remark 1.* Let $\mathfrak{R} = (\mathbb{B}, \Gamma, \delta)$ be a 1-biased reset and let $\Gamma_1 = \{a \in \Gamma | \overline{a}^{\mathfrak{R}} \equiv 1\}$. Then[4] $\mathcal{L}(\mathfrak{R}) = \{\emptyset, \Gamma^*, \bigcup_{a\in\Gamma_1} \Gamma^* a \Gamma^*, \bigcap_{a\in\Gamma_1} \overline{\Gamma^* a \Gamma^*}\}$. In particular all languages from $\mathcal{L}(\mathfrak{R})$ are piecewise testable.

We now investigate the languages recognized by cascade products of biased resets:

**Lemma 1.** *Let $\mathfrak{A} = (Q, \Gamma, \delta^{\mathfrak{A}})$ be a semiautomaton and let $\mathfrak{R} = (\mathbb{B}, \Gamma \times Q, \delta^{\mathfrak{R}})$ be a 1-biased reset. Then every language recognized by $\mathfrak{A} * \mathfrak{R}$ is a finite union of languages of one of the following three forms (for suitable $p_0, p_F \in Q$):*

*(1)* $L \in \mathcal{L}(\mathfrak{A})$

*(2)* $L = \bigcup_{(a,q)\in S_1} L(\mathfrak{A}, p_0, \{q\}) \cdot a \cdot L(\mathfrak{A}, \overline{a}^{\mathfrak{A}}(q), \{p_F\})$

*(3)* $L = \left(\overline{\bigcup_{(a,q)\in S_1} L(\mathfrak{A}, p_0, \{q\}) \cdot a \cdot \Gamma^*}\right) \cap L(\mathfrak{A}, p_0, \{p_F\})$

*where* $S_1 = \{(a,q) \in \Gamma \times Q \mid \overline{(a,q)}^{\mathfrak{R}} \equiv 1\}$.

The proof of this lemma is not difficult, but omitted due to space constraints. The reader is referred to [5].

**Lemma 2.** *If $\mathfrak{A} := \mathfrak{R}_1 * \cdots * \mathfrak{R}_n$ is a strictly locally 1-triggered cascade product of 1-biased resets $\mathfrak{R}_i = (\mathbb{B}, \Gamma \times \mathbb{B}^{i-1}, \delta^{\mathfrak{R}_i})$, $i = 1,\dots,n$, then every language recognized by $\mathfrak{A}$ with accepting states $\{(x_1,\dots,x_n) | x_n = 1\} \subseteq \mathbb{B}^n$ is a finite union of languages of the form $\Gamma^* a_1 \Gamma^* \cdots \Gamma^* a_m \Gamma^*$, $m \in \mathbb{N}_0$.*

The proof of this lemma is by induction on $n$ and uses Lemma 1 and Remark 1. For details the reader is again referred to [5].

This enables us to complete the proof of Theorem 3. Let $\mathfrak{A}$ be a locally 1-triggered cascade product of 1-biased resets. Then $\mathfrak{A}$ is a direct product of strictly

[3] We use the convention $\Gamma \times \mathbb{B}^0 = \Gamma$ and accordingly $(a,(x_1,\dots,x_{i-1})) = a$ if $i = 1$.

[4] Given a language $L \subseteq \Gamma^*$ we denote the complement language by $\overline{L} = \Gamma^* \setminus L$.

locally 1-triggered cascade products of 1-biased resets. Therefore all languages from $\mathcal{L}(\mathfrak{A})$ are Boolean combinations of the languages recognized by strictly locally 1-triggered products. Since the piecewise testable languages form a Boolean algebra, it is sufficient to show that all strictly locally 1-triggered cascade products of 1-biased resets recognize only piecewise testable languages.

Hence assume that $\mathfrak{A} := \mathfrak{R}_1 * \cdots * \mathfrak{R}_n$ is strictly locally 1-triggered. Let $q_0 \in \mathbb{B}^n$ and $F = \{(f_{1,1}, \ldots, f_{1,n}), \ldots, (f_{r,1}, \ldots, f_{r,n})\}$. Since $L(\mathfrak{A}, q_0, F) = \bigcup_{i=1}^{r} L(\mathfrak{A}, q_0, \{(f_{i,1}, \ldots, f_{i,n})\})$ we may assume that $F = \{(f_1, \ldots, f_n)\}$. Suppose $q_0 = (q_{0,1}, \ldots, q_{0,n})$ and let $q_{0,i} = 1$. Then, because $\mathfrak{A}$ is strictly locally 1-triggered, $L = L(\mathfrak{A}, q_0, F)$ is the intersection of the languages $J = L(\mathfrak{R}_1 * \cdots * \mathfrak{R}_i, (q_{0,1}, \ldots, q_{0,i}), \{(f_1, \ldots, f_i)\})$ and $K = L(\tilde{\mathfrak{R}}_{i+1} * \mathfrak{R}_{i+2} * \cdots * \mathfrak{R}_n, (q_{0,i+1}, \ldots, q_{0,n}), \{(f_{i+1}, \ldots, f_n)\})$ where $\tilde{\mathfrak{R}}_{i+1}$ is obtained from $\mathfrak{R}_{i+1}$ by treating all inputs $a \in \Gamma$ as $(a, (0, \ldots, 0, 1))$. Since both resulting products are again strictly locally 1-triggered, we may assume that $q_0 = (0, \ldots, 0)$.

Now assume that $f_n = 1$. Then, since $\mathfrak{A}$ is strictly locally 1-triggered and since all resets are 1-biased, we have $f_1 = \ldots = f_n = 1$ or the language $L = L(\mathfrak{A}, q_0, \{(f_1, \ldots, f_n)\})$ is empty. Hence $L(\mathfrak{A}, q_0, \{(x_1, \ldots, x_n) | x_n = 1\}) = L$, since $(1, \ldots, 1)$ is the only state with $x_n = 1$ reachable from $q_0$. By Lemma 2 we see that $L$ is a finite union of languages of the form $\Gamma^* a_1 \Gamma^* \cdots \Gamma^* a_r \Gamma^*$.

If $f_n = 0$ we pick $i \in \{1, \ldots, n-1\}$ maximal (if it exists) with $f_i = 1$. If no such $i$ exists, then clearly $L = \overline{L(\mathfrak{R}_1, 0, \{1\})}$, which is piecewise testable. Hence we assume such an index $i$ exists. Since $\mathfrak{A}$ is strictly locally 1-triggered we see that $f_1 = \cdots = f_i = 1$, since otherwise $(f_1, \ldots, f_n)$ is again unreachable from $q_0$. Furthermore we must have $f_{i+2} = \cdots = f_n = 0$ for the same reason. This implies that $\mathfrak{R}_{i+2}, \ldots, \mathfrak{R}_n$ are irrelevant to the acceptance behavior of $(\mathfrak{A}, q_0, (f_1, \ldots, f_n))$. Thus we may assume that $i = n - 1$.

Using the results from the case $f_n = 1$ we get that $K := L(\mathfrak{R}_1 * \cdots * \mathfrak{R}_{n-1}, (0, \ldots, 0), \{(1, \ldots, 1)\})$ is a finite union of languages $\Gamma^* a_1 \Gamma^* \cdots \Gamma^* a_r \Gamma^*$. Now by Lemma 1 $L$ is a finite union of languages of the form $\overline{K \sigma \Gamma^*} \cap K$. Since the piecewise testable languages are closed under the Boolean operations this concludes the proof.

## 4 Commutative Languages

In this section we embed the well known results on commutative languages into our framework. We first recall the definition. For $w = a_1 \cdots a_n \in \Gamma^*$ let $\mathrm{Perm}(w) = \{a_{\pi(1)} \cdots a_{\pi(n)} | \pi \in S_n\}$, where $S_n$ denotes the symmetric group on $n$ points. A language $L$ is *commutative* if $\mathrm{Perm}(w) \subseteq L$ for every $w \in L$. This is evidently the case iff $M(L)$ is commutative iff $\mathfrak{A}_L$ is commutative. Recall that a semiautomaton is commutative if $\overline{w}^{\mathfrak{A}}(p) = q$ implies $\overline{v}^{\mathfrak{A}}(p) = q$ for all $v \in \mathrm{Perm}(w)$.

**Definition 2.** *Let $L \subseteq \Gamma^*$ be regular.*

*(a) If there exists $N \subseteq \mathbb{N}_0$ and $a_0 \in \Gamma$ with $L = \{w \in \Gamma^* \mid |w|_{a_0} \in N\}$, then $L$ is called* 1-semilinear (with respect to $a_0$)[5] [6].

*(b) A semiautomaton $\mathfrak{A}$, such that there exists a letter $a_0 \in \Gamma$ with $\overline{a_0}^{\mathfrak{A}} \neq \mathrm{id}$ and $\overline{a}^{\mathfrak{A}} = \mathrm{id}$ for all $a_0 \neq a \in \Gamma$ is called* one letter automaton (OLA) (with respect to $a_0$).

1-semilinear languages are commutative. Clearly languages accepted by OLA are 1-semilinear. Furthermore 1-semilinear languages yield canonical DFA, which are OLA (for the proof one can use, for instance, Moore's minimization algorithm). A cascade product, such that the automaton it defines is an OLA, is a *one letter cascade product*. We recall that a language is commutative iff it is a Boolean combination of 1-semilinear languages (for details the reader is referred to [10]).

We now turn towards characterizing OLA by their Krohn-Rhodes decompositions. We first observe that minimal OLA (i.e. those that are canonical DFA for some (1-semilinear) language $L$) have a very simple form. If $\mathfrak{A}_L = (Q, \Gamma, q_0, \delta, F)$ is the minimal DFA for a 1-semilinear language $L$ (with respect to $a \in \Gamma$) then there exist $i < j$ minimal, such that $\overline{w_i}^{\mathfrak{A}_L}(q_0) = \overline{w_j}^{\mathfrak{A}_L}(q_0)$, where $w_k = a^k$ for $k \in \mathbb{N}_0$. Since $\mathfrak{A}_L$ is an OLA w.r.t. $a$ all states from $Q$ occur in the sequence $\overline{w_0}^{\mathfrak{A}_L}(q_0), \overline{w_1}^{\mathfrak{A}_L}(q_0), \ldots, \overline{w_{j-1}}^{\mathfrak{A}_L}(q_0)$ ($\mathfrak{A}_L$ is minimal). Set $q_k := \overline{w_k}^{\mathfrak{A}_L}(q_0)$. Then we obtain two disjoint sets $Q_{tail} = \{q_0, \ldots, q_{i-1}\}$ and $Q_{loop} = \{q_i, \ldots, q_{j-1}\}$.

**Lemma 3.** *Let $\mathfrak{A}_L$ be an OLA. Then $\mathfrak{A}_L \leq \mathfrak{R}_1 * \cdots * \mathfrak{R}_i \times \mathfrak{C}$, where $\mathfrak{R}_1 * \cdots * \mathfrak{R}_i$ is a one letter cascade product of 1-biased resets and $\mathfrak{C}$ is a cyclic one letter grouplike automaton of order $j - i$ (where $i < j$ are as above).*

*Proof.* Choose $i$ and $j$ as in the previous paragraph. For $\mathfrak{R}_1 * \cdots * \mathfrak{R}_i$ we pick the product recognizing $(\Gamma^* a \Gamma^*)^i$ as constructed in Sec. 3. We define $\mathfrak{C} = (Q_{loop}, \Gamma, \delta')$ by $\delta'(x) = \overline{x}^{\mathfrak{A}_L}|_{Q_{loop}}$ for $x \in \Gamma$. Pick $r \in Q_{loop}$ such that $\overline{w_i}^{\mathfrak{C}}(r) = q_i$, i.e. $r$ is chosen such that after seeing $i$ $a$'s we end up in the first state of $Q_{loop}$ visited when starting from $q_0$. Define $A \subseteq \mathbb{B}^i \times Q_{loop}$ by starting out from $(0, \ldots, 0, r)$ and adding all states reachable in $\mathfrak{R}_1 * \cdots * \mathfrak{R}_i \times \mathfrak{C}$. Then $A$ defines a subsemiautomaton. Notice that for $(x_1, \ldots, x_i, q) \in A$ there exists $0 \leq k \leq i$ such that $x_1 = \cdots = x_k = 1$ and $x_{k+1} = \cdots = x_i = 0$. Denote this integer $k$ by $\max(x_1, \ldots, x_i)$. Then we define $\psi : A \to Q$ by $(x_1, \ldots, x_i, q) \mapsto q_{\max(x_1, \ldots, x_i)}$ if $\max(x_1, \ldots, x_i) < i$ and $q$ otherwise. Is is left to the reader to verify that $\psi$ is a surjective homomorphism onto $\mathfrak{A}_L$. □

The grouplike automaton $\mathfrak{C}$ in the lemma above need not be simple. Hence we decompose $\mathfrak{C}$ further in order to arrive at a Krohn-Rhodes decomposition. It is well known that for every finite cyclic group $C$ we have $C \cong \mathbb{Z}/m\mathbb{Z} \cong \times_{i=1}^{s} \mathbb{Z}/m_i\mathbb{Z}$ where $|C| = m = \prod_{i=1}^{s} m_i$ and the $m_i$ are pairwise coprime prime powers. We will give an automaton theoretic equivalent of this fact. Denote by $C_n$ the group $(\mathbb{Z}/n\mathbb{Z}, +)$, $n \in \mathbb{N}$, and denote the $n$-class of an integer $i$ by $[i]_n$. Returning to

[5] The name results from the fact that the Parikh image of such a language is determined by just one dimension.

[6] Here $|w|_a$ denotes the number of occurrences of the letter $a \in \Gamma$ in $w$.

the grouplike automaton from the previous lemma, let the order of $\mathfrak{C}$ be $m$. Define $\mathfrak{H} = (C_m, \Gamma, \delta^{\mathfrak{H}})$ where $\overline{x}^{\mathfrak{H}}([i]_m) = [i+1]_m$ if $x = a$ and $[i]_m$ otherwise, $x \in \Gamma$. Evidently $\mathfrak{C} \cong \mathfrak{H}$ and so in particular $\mathfrak{C} \leq \mathfrak{H}$. Define $\mathfrak{C}_i = (C_{m_i}, \Gamma, \delta^{\mathfrak{C}_i})$ in the same way as $\mathfrak{H}$ (that is for $x \in \Gamma$ let $\overline{x}^{\mathfrak{C}_i}([k]_{m_i}) = [k+1]_{m_i}$ if $x = a$ and $[k]_{m_i}$ otherwise). Then evidently all $\mathfrak{C}_i$ are one letter cyclic grouplike automata and we have $\mathfrak{H} \cong \mathfrak{C}_1 \times \cdots \times \mathfrak{C}_s$. For the proof of this statement, one uses the (group) isomorphism obtained from classical group theory and verifies that it is also a homomorphism of automata. It remains to decompose grouplike one letter automata of the form $\mathfrak{C} = (C_{p^k}, \Gamma, \delta)$ for some prime $p$ and some $k \in \mathbb{N}$.

**Lemma 4.** *$\mathfrak{C} \leq (C_p, \Gamma, \delta^1) * \cdots * (C_p, \Gamma \times C_p^{k-1}, \delta^k)$, for cyclic grouplike automata $(C_p, \Gamma \times C_p^{i-1}, \delta^i)$, $1 \leq i \leq k$. The cascade product defines an OLA.*

*Proof.* The proof is by induction on $k$. If $k = 1$ there is nothing to show. For the induction step we define $\mathfrak{D} = (C_p, \Gamma, \delta^{\mathfrak{D}})$ by $\overline{x}^{\mathfrak{D}}([i]_p) = [i+1]_p$ if $x = a \in \Gamma$ and $[i]_p$ otherwise, $x \in \Gamma$. Then define $\mathfrak{H} = (C_{p^{k-1}}, \Gamma \times C_p, \delta^{\mathfrak{H}})$ by $\overline{(a, [p-1]_p)}^{\mathfrak{H}}([i]_{p^{k-1}}) = [i+1]_{p^{k-1}}$ and $\overline{(x, [r]_p)}^{\mathfrak{H}}([i]_{p^{k-1}}) = [i]_{p^{k-1}}$ for all $(x, [r]_p) \neq (a, [p-1]_p)$. Observe that $\mathfrak{D} * \mathfrak{H}$ is an OLA. By the induction hypothesis and Theorem 1 we are done if we show that $\mathfrak{C} \leq \mathfrak{D} * \mathfrak{H}$. To this end define $\varphi : C_p \times C_{p^{k-1}} \to C_{p^k}$ by $\varphi([i]_p, [j]_{p^{k-1}}) = [(i \mod p) + j \cdot p]_{p^k}$. This mapping is well-defined (as one easily verifies) and is a homomorphism of semiautomata. We only treat the case of the letter $a$, the case of the remaining letters being trivial. We have $\overline{a}^{\mathfrak{C}}(\varphi([p-1]_p, [j]_{p^{k-1}})) = [(p-1+j \cdot p)+1]_{p^k} = \varphi([0]_p, [j+1]_{p^{k-1}})$. If $0 \leq i < p-1$, then $\overline{a}^{\mathfrak{C}}(\varphi([i]_p, [j]_{p^{k-1}})) = [i + j \cdot p + 1]_{p^k} = \varphi([i+1]_p, [j]_{p^{k-1}})$. $\square$

In summary, we have shown:

**Theorem 4.** *Let $L \subseteq \Gamma^*$. The following are equivalent:*

*(1) $L$ is commutative*
*(2) $\mathfrak{A}_L \leq \left(\times_{i=1}^{k} \mathfrak{R}_{i,1} * \cdots * \mathfrak{R}_{i,n_i}\right) \times \left(\times_{i=1}^{r} \mathfrak{C}_{i,1} * \cdots * \mathfrak{C}_{i,m_i}\right)$, where all cascade products define one letter automata, all resets are biased, all grouplike automata are cyclic of prime order and the orders of $\mathfrak{C}_{\mathrm{i,j}}$ and $\mathfrak{C}_{i',j'}$ are equal iff $i = i'$.*

## 5 The Scope of Cascade Products

In Definition 1 we introduced the scope of a cascade product. Notice this definition was only stated for cascade products consisting of resets. We are therefore only dealing with star-free languages. We now want to use this notion to investigate language classes: Given a class $\mathcal{C}$ of star-free languages (e.g. piecewise testable languages, $\mathcal{R}$-trivial languages etc.), what can be said about the scope of a cascade product recognizing the languages from $\mathcal{C}$? If there exists $k \in \mathbb{N}_0$, such that every $L \in \mathcal{C}$ is recognized by a cascade product of scope at most $k$, then we say $\mathcal{C}$ has *constant scope* or has *scope $k$*. From Theorem 3 we can deduce:

**Proposition 2.** *The class of piecewise testable languages has scope 1.*

Recall that a language is $\mathcal{R}$-trivial if its syntactic monoid $M(L)$ is $\mathcal{R}$-trivial, i.e. if $mM(L) = nM(L)$ iff $m = n$ for all $m, n \in M(L)$. We have:

**Theorem 5.** *The class of $\mathcal{R}$-trivial languages has scope 2.*

The idea of the proof is to decompose an $\mathcal{R}$-trivial language $L$ into a union of left-deterministic products[7] and then to show how to cover the minimal automata for such products by a scope 2 product of biased resets. For a detailed proof, which we omit due to space constraints, the reader is referred to [5].

## 6 Cascade Products and Dot-Depth

In this section we will make extensive use of first order logic. We will be using formulas from the logic $\mathsf{FO}[\min, \max, (P_a)_{a\in\Gamma}, <, S]$. Such formulas will be interpreted in *word models*: Let $w = b_1 \cdots b_m \in \Gamma^*$. Then the model associated with $w$ is denoted $\underline{w} = (\{1, \dots, m\}, 1, m, (\{i | b_i = a\})_{a\in\Gamma}, <, S)$ where $S(x, y)$ iff $y = x + 1$ and $<$ is the usual order on natural numbers. If $\varphi$ is a sentence we write $L(\varphi) = \{w \in \Gamma^* | \ \underline{w} \models \varphi\}$ for the language *specified* or *accepted* by $\varphi$. If $\varphi(\bar{x})$ has free variables $\bar{x} = (x_1, \dots, x_n)$ then we write $(\underline{w}, \bar{k}) \models \varphi(\bar{x})$ for $\bar{k} = (k_1, \dots, k_n)$ if $\varphi$ holds in $\underline{w}$ with $x_i$ interpreted by $k_i$.

As usual we denote by $\Sigma_n$ the set of $\mathsf{FO}$-formulas, which are equivalent to a formula in prenex normal form with $n$ quantifier alternations beginning with a block of existential quantifiers, e.g. $\exists\bar{x}_1\forall\bar{x}_2 \cdots Q_n\bar{x}_n\varphi(\bar{x}_1, \dots, \bar{x}_n)$ where $\varphi$ is quantifier free and $Q_n$ is existential iff $n$ odd. We then define $\Pi_n$ to be the set of formulas, the negation of which is in $\Sigma_n$ and we set $\Delta_n = \Sigma_n \cap \Pi_n$. Given a set of formulas $\Phi$ we define $\mathsf{BC}(\Phi)$ to be the set of all Boolean combinations of formulas in $\Phi$. Immediately from the definitions one gets $\mathsf{BC}(\Sigma_n) = \mathsf{BC}(\Pi_n) \subseteq \Delta_{n+1}$ and $\mathsf{BC}(\Delta_n) = \Delta_n$ for all $n \in \mathbb{N}$. Also, we recall that disjunctions and conjunction of $\Sigma_n$ (resp. $\Pi_n$) formulas is again a $\Sigma_n$ (resp. $\Pi_n$) formula. Given a set $\Phi$ of formulas, we often write $L \in \Phi$ if $L = L(\varphi)$ for some sentence $\varphi \in \Phi$.

We now recall the *dot-depth* of star-free languages. To this end let $\mathcal{B}_0$ be the set of all finite and co-finite languages. For $n \in \mathbb{N}_0$ define $\mathcal{B}_{n+1}$ to be the set of all Boolean combinations of languages $L_1a_1L_2 \cdots L_{n-1}a_{n-1}L_n$ where $L_1, \dots, L_n \in \mathcal{B}_n$ and $a_1, \dots, a_n \in \Gamma$. One can show that $\bigcup_{n\in\mathbb{N}_0} \mathcal{B}_n$ is the set of star-free languages[8]. The *dot-depth* of a language $L$ is the number $n \in \mathbb{N}$, such that $L \in \mathcal{B}_n \setminus \mathcal{B}_{n-1}$ (or 0 if $L \in \mathcal{B}_0$). The dot-depth is intimately tied to logic:

**Theorem 6 (Thomas, [16]).** *$L \in \mathcal{B}_n$ iff $L \in \mathsf{BC}(\Sigma_n)$ for $n \in \mathbb{N}$.*

Notice the theorem makes no statement about level 0. The following theorem is due to Brzozowski and Cohen. In its original formulation it did not make any references to logic. We give an alternative proof using logic and can thereby place the languages more precisely within a given level of the hierarchy[9]:

[7] Recall a product $\Gamma_0a_1\Gamma_1a_2\Gamma_2 \cdots a_n\Gamma_n$ is *left-deterministic* if for every $i = 1, \dots, n$ we have $a_i \notin \Gamma_{i-1}$ (see [1, 9] for details).

[8] See [2, 9, 10] for details.

[9] The result from [2] yields $L \in \mathcal{B}_{n+1} = \mathsf{BC}(\Sigma_{n+1})$, which is a superset of $\Delta_{n+1}$.

**Theorem 7 (Cohen, Brzozowski, [2]).** *Let $L$ be recognized by a cascade product $\mathfrak{R}_1 * \cdots * \mathfrak{R}_n$ of $n$ resets. Then $L \in \Delta_{n+1}$ and so has dot-depth at most $n+1$.*

*Proof.* The proof is by induction on $n$. The induction start is left to the reader. For the induction step write $\mathfrak{A} = \mathfrak{R}_1 * \cdots * \mathfrak{R}_{n-1} = (Q, \Gamma, \delta^{\mathfrak{A}})$. Then $\mathcal{L}(\mathfrak{A}) \subseteq \Delta_n$ by the induction hypothesis. Let $L(q,b) = L(\mathfrak{A} * \mathfrak{R}_n, (q_0, b_0), \{(q,b)\})$. Then every language in $\mathcal{L}(\mathfrak{A} * \mathfrak{R}_n)$ is a union of languages of this form. It is sufficient to show $L(q,b) \in \Sigma_{n+1}$ for all states $(q,b)$. The claim follows from the fact that $\overline{L(q,b)} \in \Sigma_{n+1}$ as well, hence $L(q,b) \in \Sigma_{n+1} \cap \Pi_{n+1} = \Delta_{n+1}$.

We pick a formula $\varphi_q(x) \in \Delta_n$, such that for $w = a_1 \cdots a_m \in \Gamma^*$ we have $(\underline{w}, k) \models \varphi_q(x)$ iff $a_1 \cdots a_k \in L(q) = L(\mathfrak{A}, q_0, \{q\})$ for $q \in Q$. Denote by $\Gamma^{(i)} \subseteq \Gamma \times Q$ the set of inputs inducing the constant $i$-mapping in $\mathfrak{R}_n$ for $i \in \mathbb{B}$. Then $L(q,b)$ is specified by $\varphi_q(\max) \wedge \exists x \exists z_1 \forall y \forall z_2 \Big( \bigvee_{(a,q') \in \Gamma^{(b)}} \varphi_{q'}(x) \wedge P_a(z_1) \wedge S(x, z_1) \wedge \Big( (S(y, z_2) \wedge y > x) \rightarrow \bigwedge_{(a,q') \in \Gamma^{(1-b)}} (\neg \varphi_{q'}(y) \vee \neg P_a(z_2)) \Big) \Big)$, which is in $\Sigma_{n+1}$ since $\varphi_q \in \Delta_n$ for all $q \in Q$. ☐

We will now refine the result from Theorem 7. However, before moving on, we recall from [1] that the state set $Q$ of a cascade product of biased resets is *partially ordered*, i.e. there exists a partial order $\preceq$ on $Q$, such that for all $a \in \Gamma$ and all $q \in Q$ we have $q \preceq \overline{a}(q)$. We note that we can extend this partial order, to a total order, which is still compatible with the transitions in the way just outlined. We will say the state set is *ordered*.

**Lemma 5.** *Let $\mathfrak{A} = (Q, \Gamma, \delta^{\mathfrak{A}})$ be a semiautomaton, such that $\mathcal{L}(\mathfrak{A}) \subseteq \Delta_n$ for some $n \in \mathbb{N}$. Then $\mathcal{L}(\mathfrak{A} * \mathfrak{R}_1 * \cdots * \mathfrak{R}_k) \subseteq \Delta_{n+1}$, where $\mathfrak{R}_i$ is a biased reset for $i = 1, \ldots, k$, $k \in \mathbb{N}$.*

*Proof.* Let $S = \{1, \ldots, r\}$ be the state space of $\mathfrak{D} = \mathfrak{R}_1 * \cdots * \mathfrak{R}_k$. We may assume the compatible order on $S$ to coincides with $\leq$ (the usual order on $\mathbb{N}$). Denote by $\Gamma_{i,j} \subseteq \Gamma \times Q$ the set of inputs which map $i$ to $j$. Notice that in a run $\mathfrak{D}$ can change its state at most $r-1$ times.

Write $\mathfrak{B} := \mathfrak{A} * \mathfrak{D}$. Let $L = L(\mathfrak{B}, (q_0, i_0), \{(q_F, f)\})$, where $q_0, q_F \in Q$ and $i_0, f \in S$. Then all languages in $\mathcal{L}(\mathfrak{B})$ are unions of languages of this form (and therefore defined by disjunctions of the corresponding formulas). For $q \in Q$ and $w = a_1 \cdots a_m \in \Gamma^*$ let $\varphi_q(x) \in \Delta_n$ be such that $(\underline{w}, t) \models \varphi_q(x)$ iff $a_1 \cdots a_t \in L(\mathfrak{A}, q_0, \{q\})$. Then clearly $L(\mathfrak{A}, q_0, \{q\}) = L(\varphi_q(\max))$. We treat only the case $i_0 \neq f$. The other case requires an adjustment term, which checks the possibility of staying in $i_0 = f$. Define $\theta$ to be

$$\bigvee_{j=1}^{r-1} \exists x_1 \cdots \exists x_j \forall y_0 \cdots \forall y_j \Big( y_0 < x_1 < y_1 \cdots < y_{j-1} < x_j < y_j \wedge \bigvee_{(i_1, \ldots, i_{j-1}) \in S^{j-1}} \Big( \psi_{i_0, i_1}(x_1) \wedge \ldots \wedge \psi_{i_{j-1}, f}(x_j) \wedge \bigwedge_{k=0}^{j-1} \bigwedge_{s \neq i_k} \neg \psi_{i_k, s}(y_k) \wedge \bigwedge_{s \neq f} \neg \psi_{f,s}(y_j) \Big) \Big)$$

where $\psi_{t,u}(x) \equiv \big(x = \min \wedge \bigvee_{(a,q_0)\in\Gamma_{t,u}} P_a(\min)\big) \vee \big(x > \min \wedge \exists y(S(y,x) \wedge \bigwedge_{(a,q)\in\Gamma_{t,u}} \varphi_q(y) \wedge P_a(x))\big)$. $\psi_{t,u}(x)$ verifies that $\mathfrak{O}$ changes to state $u$ upon reading the letter at position $x$ if $\mathfrak{O}$ was in state $t$ before. $\theta$ says that for some suitable $j$ we have precisely $j$ state changes leading from $i_0$ to $f$. Since we can replace $\exists y(S(y,x) \wedge \cdots$ by $\forall y(S(y,x) \rightarrow \cdots$, we have $\psi_{i,j} \in \Delta_n$, hence $\theta \in \Sigma_{n+1}$. Again (see proof of Theorem 7) we conclude $\theta \in \Delta_{n+1}$ (the complement language is also in $\Sigma_{n+1}$). Clearly $L = L(\theta \wedge \varphi_{q_F}(\max))$, which is a $\Delta_{n+1}$ formula. □

Hence biased resets as factors in a cascade product have a limited impact on the dot-depth. We now define the *biased reset complexity* of a cascade product. Informally, the biased reset complexity is the number of resets in a product, where every block of biased resets is counted as a single reset. For instance, indicating biased resets by a square and non-biased resets by a circle, the following product has biased reset complexity 6:

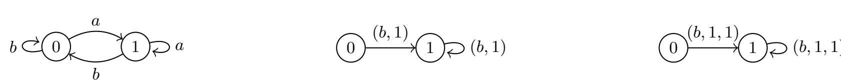
● ■ ■ ■ ● ■ ■ ● ●

More formally, let $\mathfrak{R}_1 * \cdots * \mathfrak{R}_n$ be a cascade product of resets. Let $B = \{(i,j) | \mathfrak{R}_k \text{ biased for } i \leq k \leq j \text{ and } \mathfrak{R}_{i-1}, \mathfrak{R}_{j+1} \text{ not biased}\}$. Let $m$ be the number of biased resets in the product. The biased reset complexity is $n - m + |B|$. The following theorem is now immediate:

**Theorem 8.** *Let $\mathfrak{R}_1 * \cdots * \mathfrak{R}_n$ be a cascade product of resets with biased reset complexity $k$. Then $\mathcal{L}(\mathfrak{R}_1 * \cdots * \mathfrak{R}_n) \subseteq \Delta_{k+1}$.*

The following example shows that the bound given in Theorem 8 is not tight.

*Example 1.* Consider $L := \Gamma^* a \Gamma^* b \Gamma^* a \Gamma^* b \Gamma^*$ where $\Gamma = \{a, b\}$. Then define three resets as depicted in Fig. 1. The cascade product $\mathfrak{R}_1 * \mathfrak{R}_2 * \mathfrak{R}_3$ recognizes $L$ with initial state $(0,0,0)$ and final states $\{(0,1,1),(1,1,1)\}$. Notice that Theorem 7 yields $L \in \Delta_4$, Theorem 8 yields $L \in \Delta_3$, but $L \in \Sigma_1$ as one easily verifies.




(a) $\mathfrak{R}_1$ (b) $\mathfrak{R}_2$ (c) $\mathfrak{R}_3$

**Fig. 1.** Resets for the language $L$. Inputs, which have not been specified, are assumed to induce the identity mapping.

## 7 Conclusion

Motivated by the Krohn-Rhodes Theorem we classified several classes of regular languages via their cascade decompositions. The concepts used, respectively introduced in this study, were biased resets, locally triggered cascade products, and the scope and the biased reset complexity of cascade products.

This paper gives initial results on the introduced concepts; it leads to several interesting questions left open here. For example, it should be answered whether for each $n$ there is a star-free language $L_n$ which needs at least scope $n$ in the cascade decomposition of any (minimal) automaton accepting $L_n$. As another direction for future research (connected with the final result), we mention that the biased reset complexity of decompositions of automata for star-free languages seems to deserve a closer study.

**Acknowledgments** The author thanks his supervisor Wolfgang Thomas and the reviewers of this paper for their advice and many helpful suggestions.

## References

1. Brzozowski, J.A., Fich, F.E.: Languages of J-trivial monoids. Journal of Computer and System Sciences **20**(1) (1980) 32 – 49
2. Cohen, R.S., Brzozowski, J.A.: Dot-depth of star-free events. Journal of Computer and System Sciences **5**(1) (1971) 1 – 16
3. Eilenberg, S.: Automata, Languages, and Machines: Volume A. Pure and Applied Mathematics. Elsevier, Burlington, MA (1974)
4. Eilenberg, S.: Automata, Languages, and Machines: Volume B. Pure and Applied Mathematics. Elsevier, Burlington, MA (1974)
5. Gelderie, M.: Classifying regular languages via cascade products of automata. Diploma thesis, RWTH Aachen University (2011)
6. Ginzburg, A.: Algebraic Theory of Automata. Academic Press, New York (1968)
7. Krohn, K., Rhodes, J.: Algebraic theory of machines. I. Prime decomposition theorem for finite semigroups and machines. Trans. Amer. Math. Soc **116** (1965) 450–464
8. Meyer, A.R.: A note on star-free events. J. ACM **16**(2) (1969) 220–225
9. Pin, J.E.: Varieties Of Formal Languages. Plenum Publishing Co. (1986)
10. Pin, J.E.: Syntactic semigroups. In Rozenberg, G., Salomaa, A., eds.: Handbook of Formal Languages, Vol. 1: Word, Language, Grammar. Springer-Verlag New York, Inc., New York, NY, USA (1997) 679–746
11. Schützenberger, M.P.: On finite monoids having only trivial subgroups. Information and Control **8**(2) (1965) 190 – 194
12. Simon, I.: Piecewise testable events. In: Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages, London, UK, Springer-Verlag (1975) 214–222
13. Straubing, H.: On finite J-trivial monoids. Semigroup Forum **19** (1980) 107–110
14. Straubing, H.: Finite Automata, Formal Logic, and Circuit Complexity. Birkhauser Verlag, Basel, Switzerland, Switzerland (1994)
15. Straubing, H., Thérien, D.: Partially ordered finite monoids and a theorem of I. Simon. J. Algebra **119**(2) (1988) 393–399
16. Thomas, W.: Classifying regular events in symbolic logic. Journal of Computer and System Sciences **25**(3) (1982) 360 – 376